Adaptivity and Learning in Intelligent Real-Time Systems

Jürgen Lind, Christoph G. Jung, and Christian Gerber
German Research Center for Artificial Intelligence (DFKI)
Im Stadtwald, Bau 36.1, 66123 Saarbrücken, Germany
email: {lind, jung, gerber}@dfki.de

1 Introduction

Intelligent monitoring and control is a key technique in modern real-time technology. The *abstract resource* framework provides an adaptive agent architecture equipped with a scheduler that steers the agent's computation using *internal profiles*: Utility-based meta-control trades off the cost of invested resources against the *expected performance*.

As in other approaches to meta-control, the important question of how to incorporate *external* performance feedback has however not been addressed until now: For situated agents, external profiling is a more natural way of predicting utility, because it grounds its estimations on observational feedback with respect to ultimate goals of the agent

This paper presents a control process based on abstract resources that additionally considers *confidence* into internal profiling. Confidence measures are derived from external stimuli by a multi-method learning algorithm. The approach is exemplified and evaluated at hand of the RoboCup simulation domain that provides a benchmark for adaptivity and learning in intelligent real-time systems.

2 The RoboCup Domain

The RoboCup simulation league [5] provides a benchmark for intelligent real-time multi-agent systems. In the simulation, each player is represented by a separate program that is connected via network to a central simulation server. In real-time (150ms perception; 100ms action), these programs have to command their associated virtual bodies (dashing, turning, kicking, catching) fed by noisy synthetic perception (relative positions and motions).

3 A Resource-Adapting Architecture

InteRRaP soccer agents [3] consist of a *Behavior-Based Layer* (BBL) that incorporates *concurrent* behavioral *processes* and a supervising *Local Planning Layer* (LPL) that incorporates planning processes. Our soccer agents follow the resource-adapting extension of [1] in which the LPL explicitly adapts the BBL towards rationality; the interface between LPL and BBL is represented by a special *control process*.

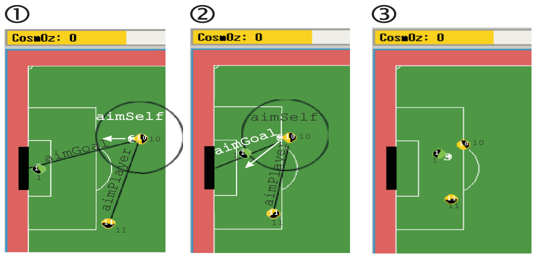

Figure 1: The RoboCup Simulation

3.1 Reactive, Concurrent Behavior

The BBL provides a tight feedback loop with the environment realized by concurrent behavioral *processes*. For example, in each of its *chunks* (cycles), the `perception` behavior senses data from a UDP datagram socket. The output percept is incorporated into the state of the BBL, the so-called *world knowledge*. Changes in the world knowledge invoke the triggering of *signals* to activate chunks of other processes, e.g., processes that produce absolute coordinates from the relative perception via triangulation.

This reactive net of processes already incorporates the whole functionality of soccer agents, i.e., the complete motivational basis for all different situations and all different player roles *at the same time*. This eventually raises conflicts because processes are unaware of all the effects of their decisions.

3.2 Deliberation as Resource Allocation

[1] introduces *abstract resources* as a discrete, quantitative representation for each of those interdependencies. Dependent processes 'consume' resources by each of their chunks and once an abstract resource is exhausted, the applying processes cannot compute anymore. In our soccer agents, for example, the limited `aim` resource (range $\{0,1\}$) restricts the concurrent and conflicting operation of the three aim processes. `aimSelf`, `aimPlayer`, and `aimGoal` apply for it; only one is able to get a hold on. `landmark` (range $\{0,1\}$), `stamina` (range $\{0,\ldots,2500\}$), `track` (range $\{0,1\}$) and finally `actuator` (range $\{0,1\}$) are further abstract resources which address the remaining conflicts in our soccer agents.

Based on those representations, we specify a simple control process that minimizes conflicts and maximizes the expected utility of a behavior schedule. In principle, this control process implements the signals that are needed to resume process activity. During each cycle, candidate signals are generated from the world knowledge. These are sorted according to the expected utility of running a respective process. The performance estimations are here produced by process built-in evaluation functions.

Figure 1 shows how these reports drive the control of player 10. In situation 1, distances to the goal and to the team mate 11 are rather large. The controller runs `aimSelf`, because there is no opponent nearby and thus dribbling has the best evaluation and utility. `aim` is then exhausted; both `aimGoal` and `aimPlayer` are suspended. In situation 2, we recognize the opponent's goalie having approached the offense player which penalizes dribbling. Since the goal is closer than the team mate, `aimGoal` is scheduled.

4 Learning Confidence into Process Evaluation

Although on-line profiling by self-evaluation performed quite successfully in the RoboCup-98 competition, situation 1 in Figure 1 reveals its drawback: The decision of player 10 to prefer aiming at the goal over aiming to the team mate has certainly not been optimal, because the opponent's goalie 1 can intercept the ball easily while number 11 is in a better position.

We introduce the idea of adjustable *confidence* measures to overcome this limitation. Confidence into performance profiles represents the agent's experience with running a particular process in a particular situation and is stored in a lookup table (*memory-based reasoning* [6], Section 4.2). Confidence measures steer the internal control and are acquired by applying a reinforcement learning [4] algorithm `confidlearn` to external feedback at runtime.

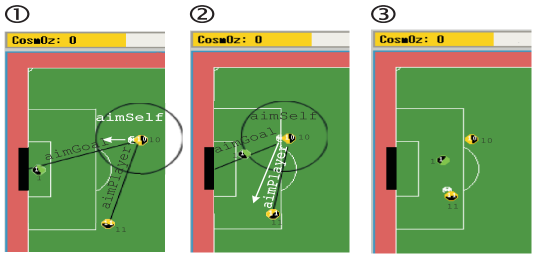

Figure 2: Adjusted Confidence into `aimGoal`

4.1 The Bucket Brigade: Updating Confidence Measures

The first problem for `confidlearn` is the fact that it is not used in the typical way for predicting the value of actions, but for learning to estimate computations. Behavior processes, such as `aimGoal`, `aimPlayer`, and `aimSelf` do not have external, but mainly internal effects.

The bucket brigade algorithm [2] was originally introduced in the context of rule based systems. Each rule is assigned a particular priority that is increased every time the rule was active. At the same time, the receiving rule must "pay" a particular amount of its priority to the rules that where active in the prior cycle because these rules created the context that allowed the paying rule to become active. Identifying rules with process chunks, a similar mechanism is applicable to confidences.

We extend the *bucket brigade algorithm* [2] not only to distribute positive and negative feedback among the currently active processes and the processes that were active in the prior cycle, but also among the processes that were active at times $t-2, \ldots, t-m$.

4.2 Memory-Based Reasoning: Matching Experiences

For situated agents, such as our soccer team, situation recognition is a difficult problem because of the complex environment of the agent. We propose a memory-based reasoning approach [6] which bases on the observation that only few of all possible situations really occur. Experiences of the agent are explicitly stored as so-called *prototypical situations*. These are fully specified situations and a *matching* procedure decides about their similarity. Every new experience is matched against all prototypical situations and whenever a similar one is found, its confidence is updated. If no similar situation is found, the current experience is stored as a new prototypical situation. The advantage of the incremental approach is not to waste memory for unexperienced situations.

4.3 Multi-Method Confidence Learning

Algorithm `confidlearn`, is the result of integrating the two learning methods into a unified technique: whenever the agent receives external feedback at time t, the experience base of the agent is sought for a prototypical situation that matches the current situation of the agent given by the contents of the world knowledge at time $t-1$. If no matching prototypical situation is found, a new column in the confidence matrix is initialized. The external feedback is then equally distributed among the processes (behaviors) that were active at $t-1$ because they caused the feedback to occur. In a second loop of the algorithm the behaviors that were active at times $t-2, \ldots, t-m-1$ in the previous prototypical situation receive a discounted fraction of the feedback of their successors. The paid amount is subtracted from the successors' confidence in prototypical situation.

Applying `confidlearn` to the described RoboCup situation results in suppressing the `aimGoal` process and supporting the `aimPlayer` process. The ball is passed to teammate 11 which can easily score (see Figure 2).

Acknowledgments This work has been supported by the "Deutsche Forschungsgemeinschaft" (DFG) and Siemens AG.

References

[1] C. Gerber and C. G. Jung. Resource management for boundedly optimal agent societies. In *Proceedings of the ECAI´98 Workshop on Monitoring and Control of Real-Time Intelligent Systems*, pages 23–28, 1998.

[2] J. H. Holland. Properties of the bucket brigade algorithm. In J. J. Grefenstette, editor, *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, 1985.

[3] C. G. Jung. Experimenting with layered, resource-adapting agents in the robocup simulation. In *Proc. of the ROBOCUP'98 Workshop*, 1998.

[4] Leslie Pack Kaelbling and Michael L. Littman. Reinforcement learning: A survey. *Journal of Artificial Intelligence*, 4:237 – 285, 1996.

[5] H. Kitano, M. Tambe, P. Stone, M. Veloso, S. Coradeschi, E. Osawa, H. Matsubara, I. Noda, and M. Asada. The robocup synthetic agent challenge. In *RoboCup-97: Robot Soccer World Cup I*, volume 1395 of *Lecture Notes in Artificial Intelligence*, pages 62–73. Springer, 1998.

[6] Stanfill and Waltz. Towards memory-based reasoning. *Communications of the ACM*, 29(12), 1986.